Increasing the Efficiency of Support Vector Machine by Simplifying the Shape of Separation Hypersurface

Yiqiang Zhan[1,2,3] and Dinggang Shen[2,3]

[1] Dept. of Computer Science, The Johns Hopkins University, Baltimore, MD
[2] Center for Computer-Integrated Surgical Systems and Technology, The Johns Hopkins University, Baltimore, MD
[3] Sect. of Biomedical Image Analysis, Dept. of Radiology, University of Pennsylvania, Philadelphia, PA
{Yiqiang.Zhan, Dinggang.Shen}@uphs.upenn.edu

Abstract. This paper presents a four-step training method for increasing the efficiency of support vector machine (SVM) by simplifying the shape of separation hypersurface. First, a SVM is initially trained by all the training samples, thereby producing a number of support vectors. Second, the support vectors, which make the hypersurface highly convoluted, are excluded from the training set. Third, the SVM is re-trained only by the remaining samples in the training set. Finally, the complexity of the trained SVM is further reduced by approximating the separation hypersurface with a subset of the support vectors. Compared to the initially trained SVM by all samples, the efficiency of the finally-trained SVM is highly improved, without system degradation.

1 Introduction

Support vector machine (SVM) is a statistical classification method proposed by Vapnik in 1995 [1], and it is one of the most interesting developments in classifier design. Given m labeled training samples, i.e. $\{(\vec{x}_i, y_i) \mid \vec{x}_i \in R^n, y_i \in \{-1,1\}, i = 1 \cdots m\}$, SVM is able to generate a separation hypersurface that has maximum generalization ability. In application, a testing sample $\vec{x}$ is classified by calculating its distance to the hypersurface:

$$d(\vec{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \tag{1}$$

where α_i and b are the parameters determined by SVM's learning algorithm, and $K(\vec{x}_i, \vec{x})$ is the kernel function. Those samples $\vec{x}_i$ with nonzero parameters α_i are called "support vectors" (SVs).

SVM is widely used in different application as a powerful classifier. However, SVM usually needs a huge number of SVs to maximize the generalization ability. The huge number of SVs unavoidably increases the computational burden when classifying a new sample by calculating Eq. (1). This disadvantage thus limits the capability of SVM in the applications that require a massive number of classifications or real-time

J. Zhang, J.-H. He, and Y. Fu (Eds.): CIS 2004, LNCS 3314, pp. 732–738, 2004.

classification. Therefore, it is important to decrease the computational cost of SVM, by reducing the number of SVs.

In this paper, a novel training method is proposed to improve the efficiency of SVM classifier, by selecting appropriate training samples. Since the number of SVs determines the computational cost of SVM and are also highly related to the geometric complexity of the separation hypersurface, the basic idea of our training method is to exclude the samples that incur the separation hypersurface highly convoluted.

2 Methods

Reducing the computational cost of the SVM is equivalent to decreasing the number of the SVs. According to their positions in the feature space, SVs can be categorized into two types. The first type of SVs are the training samples that exactly locate on the margins of the separation hypersurface, i.e., $d(\vec{x}_i) = \pm 1$, as the gray circles/crosses shown in Fig 1. Their number is directly related to the shape of the separation hypersurface, i.e., the more the SVs of this type, the more convoluted the hypersurface. The second type of SVs are the training samples that locate beyond their corresponding margin, i.e., $y_i d(\vec{x}_i) < 1$, as the dashed circles/crosses shown in Fig 1. For SVM, these training samples are regarded as mis-classified samples even though some of them still locate at the correct side of the hypersurface.

SVM usually has a huge number of SVs, when the distributions of the positive and the negative training samples highly overlap with each other. This is because, (1) a large number of the first-type SVs are needed to construct a highly convoluted hypersurface, in order to separate two classes; (2) even the highly convoluted separation hypersurface has been constructed, a lot of confounding samples will be misclassified, and thus selected as the second type of SVs.

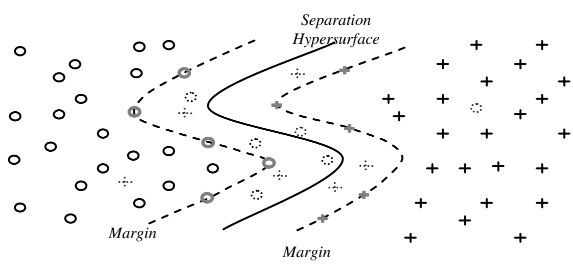


Fig. 1. Schematic explanation of the separation hypersurface, margins and SVs of SVM

Osuna have proposed an effective method to reduce the number of SVs of the trained SVM without system degradation. This method approximates the separation hypersurface with a subset of the SVs using Support Vector Regression Machine (SVRM) [4]. However, in many real applications, while SVM generates a highly convoluted separation hypersurface in the high dimensional feature space, Osuna's method still needs a large number of SVs to approximate the hypersurface. Obviously, an efficient way to further decrease the number of the SVs is to simplify the shape of the separation hypersurface, by sacrificing a very limited classification rate.

An intuitive method to simplify the shape of the hypersurface is to exclude some training samples, thereby the remaining samples are possible to be separated by a less convoluted hypersurface. However, the exclusion of training samples inevitably decreases the variety of the training set and may further influence the classification rate of the finally trained SVM. To minimize the loss of the classification rate, only the training samples that have largest contributions to the convolution of the hyper-surface are preferred to be excluded from the training set. Since the SVs determine the shape of the separation hypersurface, they are the best candidates to be excluded from the training set, in order to simplify the shape of the separation hypersurface.

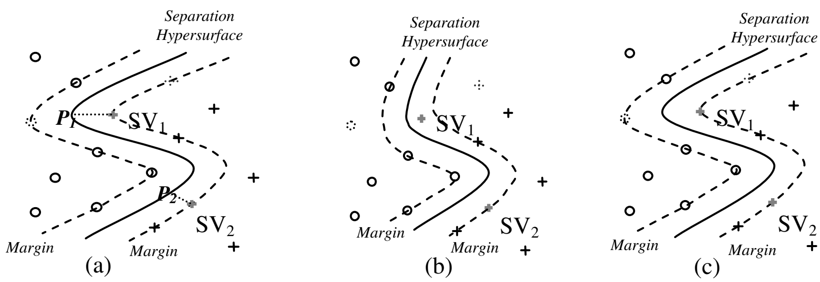


Fig. 2. Schematic explanation of how to selectively exclude the SVs from the training set, in order to effectively simplify the separation hypersurface

Excluding different sets of SVs from the training set will lead to different simplifications of the separation hypersurface. Fig 2 presents a schematic example in the 2-dimensional feature space, where we assume SVs exactly locating on the margins. As shown in Fig 2(a), SVM trained by all the samples has 10 SVs, and the separation hypersurface is convoluted. Respective exclusion of two different SVs, SV_1 and SV_2, denoted as gray crosses in Fig 2(a), will lead to two different separation hypersurfaces as shown in Figs 2(b) and 2(c), respectively. SVM in Fig 2(b) has only 7 SVs, and its hypersurface is less convoluted, after re-training SVM with all samples except SV_1, which was previously selected as a SV in Fig 2(a). Importantly, two additional samples, denoted as dashed circle/cross, were previously selected as SVs in Fig 2(a), but they are no longer selected as SVs in Fig 2(b). In contrast, SVM in Fig 2(c) still has 9 SVs, and the hypersurface is very similar to that in Fig 2(a), even SV_2, which was previously selected as a SV in Fig 2(a), has been excluded from the training set. Obviously, the computational cost of SVM in Fig 2(b) is less than that in Fig 2(c), while the correct classification rates are the same.

It is usually more effective to simplify the shape of the hypersurface by excluding the SVs, like SV_1, which contribute more to the convolution of the hypersurface. For each SV, its contribution to the convolution of hypersurface can be approximately defined as *the generalized curvature* of its projection point on the hypersurface. The projection point on the hypersurface can be located by projecting each SV to the hypersurface along the gradient of the distance function. For example, for SV_1 and SV_2 in Fig 2(a), their projection points on the hypersurface are P_1 and P_2. Obviously, the curvature of the hypersurface at point P_1 is much larger than that at point P_2, which means SV_1 has more contribution to make the hypersurface convoluted. Therefore, it is more effective to “flatten” the separation hypersurface by excluding the SVs, like SV_1, with their projection points having the larger curvatures on the hypersurface.

Based on the above idea and combined with Osuna’s method, our training method is designed to have four steps:

Step 1. Use all the training samples to train an initial SVM, resulting in l_1 SVs $\{SV_i^{\mathrm{In}}, i=1,2,...,l_1\}$ and the corresponding decision function $d_1(\vec{x})$.

Step 2. Exclude from the training set the SVs, whose projections on the hypersurface have the largest curvatures:

2a. For each SV_i^{In}, find its projection on the hypersurface, $p(SV_i^{\mathrm{In}})$, along the gradient of distance function $d_1(\vec{x})$. Then, calculate the generalized curvature of $p(SV_i^{\mathrm{In}})$ on the hypersurface, $c(SV_i^{\mathrm{In}})$.

2b. Sort SV_i^{In} in the decrease order of $c(SV_i^{\mathrm{In}})$, and exclude the top n percentage of SVs from the training set.

Step 3. Use the remaining samples to re-train the SVM, resulting in l_2 SVs $\{SV_i^{\mathrm{Re}}, i=1,2,...,l_2\}$ and the corresponding decision function $d_2(\vec{x})$. Notably, l_2 is usually less than l_1.

Step 4. Use the l_2 pairs of data points $\{SV_i^{\mathrm{Re}}, d_2(SV_i^{\mathrm{Re}})\}$ to finally train the SVRM, resulting in l_3 SVs $\{SV_i^{\mathrm{Fl}}, i=1,2,\cdots l_3\}$ and the corresponding decision function $d_3(\vec{x})$. Notably, l_3 is usually less than l_2.

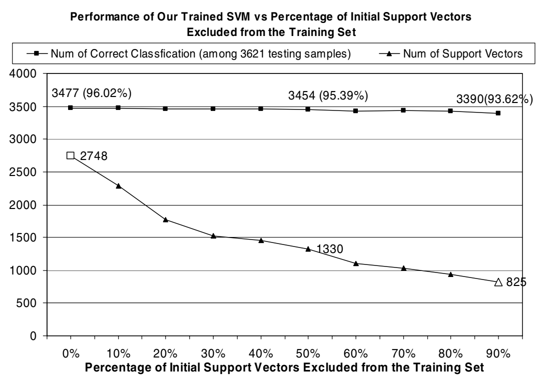
3 Experiments

In our study of 3D prostate segmentation from ultrasound images [2], SVM is used for texture-based tissue classification to differentiate prostate tissues. It is very necessary to speed up the tissue classification algorithm as the real-time segmentation is usually required in clinical applications.

The experimental data are the prostate and non-prostate samples collected from six manually labeled ultrasound images. 3621 samples from one image are used as testing samples, while 18105 samples from other five images are used as training samples. Each sample has 10 texture features, extracted by a Gabor filter bank [2].

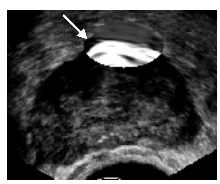
In the first experiment, we use our method to train a series of SVMs by excluding different percentages of SVs in Step 2c. As shown in Fig 3(a), after excluding 50% of initially selected SVs, the finally-trained SVM has 1330 SVs, which is only 48% of the SVs (2748) initially selected in the original SVM; but its classification rate still reaches 95.39%. Compared to 96.02% classification rate achieved by original SVM, the loss of classification rate is relatively trivial. If we want to further reduce the computational cost, we can exclude 90% of initially selected SVs from the training set. Our finally-trained SVM has only 825 SVs, which means the speed is triple, and it still has 93.62% classification rate. To further validate the effect of our trained SVM in prostate segmentation, the SVM with 825 SVs (denoted by the white triangle in Fig 3(a)) is applied to a real ultrasound image for tissue classification. As shown in Fig 3(b1-b2), the result of our trained SVM is not inferior to that of the original SVM with 2748 SVs, in terms of differentiating prostate tissues from the surrounding ones.

In the second experiment, we compare the performances of different training methods in reducing the computational cost of the finally-trained SVM and also in correctly classifying the testing samples. The five methods are implemented for comparison; they are (1) a method of slackening the training criterion by decreasing the penalty factor to errors [3]; (2) a heuristic method, which assumes the training samples distributing in a multi-variant Gaussian way, then excludes the "abnormal" training samples distant from the respective distribution centers, and finally trains a SVM only by the remaining samples; (3) a method of excluding the initially-selected SVs from the training set and then training a SVM only by the remaining samples, i.e., our proposed method without using Step 4; (4) Osuna's method [4]; (5) our proposed method. The performances of these five methods are evaluated in Fig 4(a), by the number of SVs used vs the number of correct classifications achieved. By checking the beginning curves of methods 1-5, Osuna's method is the most effective in initially reducing the number of SVs. However, to further reduce the SVs with limited sacrifice of classification rate, our method has better performance than Osuna's method.



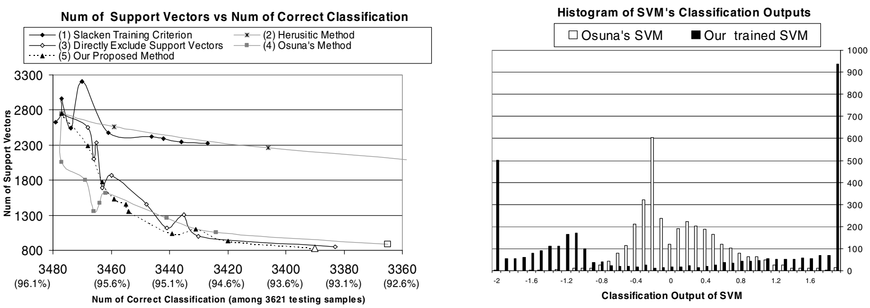
(a)

(b1)



(b2)

Fig. 3. (a) The performance of the finally-trained SVM changes with the percentages of initial SVs excluded from the training set. (b1-b2) Comparisons of tissue classification results using (b1) the original SVM with 2748 SVs and (b2) our trained SVM with 825 SVs. The tissue classification results are shown only in an ellipsoidal region



(a) (b)

Fig. 4. (a) Comparing the performances of five training methods in increasing the efficiency of SVM. (b) Histograms of classification outputs on a testing dataset respectively from our trained SVM (black bars) and Osuna's SVM (white bars)

The classification abilities of two SVMs, respectively trained by Osuna's method and our method, are further compared. The SVM trained by Osuna's method, as denoted by the white square in Fig 4(a), needs 884 SVs and its classification rate is 92.93%. The SVM trained by our method, as denoted by the white triangle in Fig 4(a), needs only 825 SVs, while its classification rate is 93.62% higher than that produced by Osuna's method. Moreover, our trained SVM actually has much better classification ability than the SVM trained by Osuna's method, once checking the histograms of their classification outputs. As shown in Fig 4(b), the classification outputs of Osuna's SVM concentrate around 0, which means the classification results of the positive and the negative samples are not widely separated. In contrast, most classification outputs of our trained SVM are either larger than 1.0 or smaller than -1.0. This experiment further proves that our training method is better in keeping the classification ability of the finally-trained SVM, after reducing many SVs.

4 Conclusion

We have presented a training method to increase the efficiency of SVM for fast classification, without system degradation. By finding that different SV has different contribution in constructing the separation hypersurface, we proposed a method to exclude the SVs that incur the separation hypersurface highly convoluted from the training set, thereby our finally trained SVM has a less number of SVs and the computational cost is reduced. Combined with Osuna's method, which using SVRM to efficiently approximate the hypersurface, our proposed method can highly increase the classification speed of the SVM, with very limited loss of classification ability. Experiments on real prostate ultrasound images demonstrate the performance of our proposed training method in discriminating the prostate tissues from other tissues. Compared to other four training methods, our proposed training method is able to generate more efficient SVMs, with better classification abilities.

References

1. Vapnik, V.N., *Statistical Learning Theory*, New York: John Wiley & Sons, (1998).
2. Zhan, Y. and Shen, D., "Automated Segmentation of 3D US Prostate Images Using Statistical Texture-Based Matching Method", *MICCAI, 2003*, Nov 16-18, Canada, (2003).
3. Burges, C.J.C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery,* Vol. 2, (1998), 121-167.
4. Osuna, E. and Girosi, F., "Reducing the run-time complexity of Support Vector Machines", *ICPR*, Brisbane, Australia, (1998).